

REMARKS

Claims 1, 10, 11, 20, 21 and 30 have been amended. No claims have been added. Claims 3, 12 and 22 have been canceled without prejudice or disclaimer. Accordingly, claims 1-2, 4-11, 13-21 and 23-30 remain pending in the application.

35 U.S.C. § 103

Claim 1 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Shoup et al., U.S. Pat. Appl. Pub. No. 2002/0147734, (hereafter "Shoup") in view of Melahn, U.S. Patent No. 6,003,042, (hereafter "Melahn"). Claims 2-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Shoup in view of Melahn, and further in view of Sawdon et al., U.S. Pat. Appl. Pub. No. 2003/0158873, (hereafter "Sawdon"). Applicant respectfully traverses theses rejections and requests reconsideration and withdrawal of the rejections for the following reasons.

Applicant's Reply to 'Response to Arguments' and Rejection of Claim 1

In Part 5 of the Office Action (Response to Arguments), the Office Action states:

In response to applicant's arguments regarding the amended claims 1 and 11, the applicant argues that "Melahn and Shoup do not teach or suggest that a first hash value is used to determine whether the original file has changed and/or a second hash value is used to determine whether the format converted file has changed.

The examiner responds that the prior art in fact teaches the feature. The Shoup reference teaches about archiving files in original and format converted format (Shoup: Paragraph 5, lines 1-10). The Melahn reference teaches about computing hash values of each existing file and for each new version of the existing file to compare if the versions match (Melahn: Column 2, line 32-43).

However, it is respectfully submitted that this response in the Office Action misses the point of Applicant's argument. Under Applicant's invention, the original and format-converted files may be archived over a long period of time. During this period of time, the disk drives on which the files are stored may deteriorate, be exposed to magnetic radiation, or the volume in which the files are contained may have been migrated to other storage devices and improperly copied, or other such scenarios may take place to corrupt or otherwise change the stored data (see, e.g., specification, page 1, lines 18-25). Thus, it is possible for one of the files to become corrupted or otherwise changed from its original state while stored in the archived storage system.

Under the invention, the storage system can check the hash value of a file at a later point in time to determine whether the file has become corrupted (see, e.g., table 4 at page 11 of the specification and accompanying text). The storage system is able to periodically read the stored files, calculate a new hash value for each file, and compare the newly-calculated hash value with the hash value originally stored in the file's inode to determine if the file has been corrupted or is otherwise changed. For example, in table 4, for "file2", the "jpg" version of file2 has changed from its

original state when it was first stored, as indicated by the "NO". Thus, the data for this file may be corrupted or otherwise changed or unreadable. Applicant's invention is able to detect this change in status, and provide notification to an administrator via table 4, which indicates that the status of the jpg version of file2 is not "OK". The administrator may then instruct creation of a new jpg version of file2, if necessary, to replace the corrupted version, such as converting from the txt or pdf version. Similarly, if the status of an original file changes, it may be recovered from a format converted file.

Melahn on the other hand teaches a space-saving method that compares an old version of a file with a later generation of a file to determine which portions of the file have been changed in the later generation so that duplicate data is not stored (see col. 2, lines 32-43). Melahn creates scripts that contain the records in the old version that do not match the new version, and then discards the old version of the file (see col. 2, lines 44-54). Thus, it is respectfully submitted that Melahn teaches nothing about determining the status of a file by comparing a recently-calculated hash value with an original hash value for the file, as in the present invention. In fact, Melahn teaches away from the present invention, since Melahn is directed to storing only a single version of a file, while Applicant's invention is directed to storing multiple versions of the same data in different file formats.

Further, Shoup fails to make up for the shortcomings in Melahn set forth above. Shoup teaches storing data files in multiple formats for archiving. Under

Shoup, a file is stored in its original format and at least one other format to provide redundancy and longevity (see e.g., paragraph 0045). However, Shoup teaches nothing about determining the status of these archived files at a later point in time by reading the files back out and comparing a newly-calculated hash value with an original hash value to determine if the archived file has been changed or corrupted.

Claim 1 has been amended to incorporate subject matter from canceled claim 3 to further clarify Applicant's invention. Claim 1 now includes that the "storage system is configured to determine whether the original file has changed or whether the at least one format converted file has changed by reading one of said files, calculating a new hash value for the read file, and comparing said new hash value with a respective one of said first hash value if said original file is read, or a corresponding second hash value if one of said format converted files is read." Neither Shoup, nor Melahn, nor any combination thereof teaches or suggests a storage system configured to achieve this. Accordingly, it is respectfully submitted that independent claim 1 is patentable over the combination of Shoup, Melahn and the other art of record whether taken singly, or in combination.

Discussion of Remaining Claims

Claim 11 has been further amended to clarify that it includes "calculating a first hash value for the original file and storing said first hash value in said original inode; calculating a second hash value for each said at least one format converted

file and storing each second hash value in the secondary inode corresponding to that format converted file". Thus, the hash value calculated for each file is stored in that file's respective inode for later use.

Sawdon teaches the use of inodes that contain file attributes and that specify the physical disk addresses of data blocks that store the actual data of the file (see, e.g., paragraph 55). Thus, Sawdon teaches nothing more than conventional inodes. Under Applicant's invention, the inode of the original file contains inode numbers of all the format converted files to which the original file was converted. Sawdon does not teach or suggest this. Further, in Applicant's invention, the original hash value for each file is calculated and stored to that file's inode. Then, at a later point in time, when the integrity of the file is checked by calculating a new hash value, the original hash value may be easily obtained from the file's inode, rather than having to search for the original hash value in a separate location. Neither Sawdon, nor Shoup, nor Melahn, nor any of the other art of record teaches or suggests this. Accordingly, claim 11 is patentable over the combination of these references.

Furthermore, claim 11 includes the limitation "using said first hash value to determine whether the original file has changed and/or using said second hash value to determine whether the format converted file has changed." As discussed above, the art of record does not teach or suggest using hash values to determine whether a file has been corrupted or otherwise changed. Accordingly, claim 11 is also

patentable over the art of record for the reasons discussed above with respect to claim 1.

Similarly, claim 21 now includes that "wherein said first hash value is stored in said first inode, and is used to determine whether the original file has changed, and wherein each said second hash value is stored in the corresponding second inode, and is used to determine whether the corresponding format converted file has changed." Thus, for the reasons discussed above with respect to claims 1 and 11, claim 21 is patentable over the art of record.

Further, it is noted that claims 7, 17 and 27 are directed to creating a status table that indicates if the archived files have been changed, and whether a format-converted file is able to reconverted to an original format file. As discussed above with respect to the rejection of claim 1, none of the prior art of record teaches or suggests this. The remaining dependent claims are also patentable over the art of record at least because they depend from allowable base claims.

Request for In-Person Interview

Applicant's undersigned attorney will contact the Examiner to request an in-person interview to discuss the foregoing remarks and amendments to the claims, and to attempt to reconcile any outstanding differences in interpretation of the prior art relative to the present invention in a more expeditious manner.

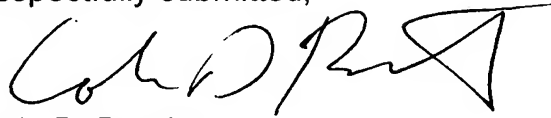
Appl. No. 10/802,852
Amendment dated September 1, 2006
Reply to Office Action of June 1, 2006

Docket No. 274.43201X00

Conclusion

In view of the foregoing, Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Colin D. Barnitz', with a stylized, sweeping flourish at the end.

Colin D. Barnitz
Registration No. 35,061

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.
1800 Diagonal Rd., Suite 370
Alexandria, Virginia 22314
(703) 684-1120
Date: September 1, 2006